

.REM 1

IDENTIFICATION

PRODUCT CODE: AC-E6820-MC
PRODUCT NAME: CXTAAD0 TA11 MODULE
PRODUCT DATE: SEPTEMBER 1978
MAINTAINER: DEC/X11 SUPPORT GROUP

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS MANUAL.

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED TO THE PURCHASER UNDER A LICENSE FOR USE ON A SINGLE COMPUTER SYSTEM AND CAN BE COPIED (WITH INCLUSION OF DIGITALS COPYRIGHT NOTICE) ONLY FOR USE IN SUCH SYSTEM, EXCEPT AS MAY OTHERWISE BE PROVIDED IN WRITING BY DIGITAL.

DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.

COPYRIGHT (C) 1974,1978 DIGITAL EQUIPMENT CORPORATION

1. ABSTRACT
.....

TAA IS AN IDMOD THAT EXERCISES EITHER OR BOTH DRIVES OF A
TAA CASSETTE UNIT.

2. REQUIREMENTS
.....

HARDWARE: TAA WITH AT LEAST ONE UNIT LOADED WITH A CASSETTE.

STORAGE: TAA REQUIRES:

- 1. DECIMAL WORDS: 518
- 2. OCTAL WORDS: 1006
- 3. OCTAL BYTES: 2014

3. PASS DEFINITION
.....

ONE PASS CONSISTS OF 70 BLOCKS WRITTEN, READ, AND CHECKED.

4. EXECUTION TIME
.....

A PASS REQUIRES APPROXIMATELY 1 MINUTE WHEN RUNNING ALONE
ON A PDP-11/05.

5. CONFIGURATION REQUIREMENTS
.....

A. DEFAULT PARAMETERS

DEVADR: 177500, VECTOR: 260, NRI: 6, DEVCNT: 2

B. REQUIRED PARAMETERS

NONE

6. DEVICE/OPTION SETUP

EACH CASSETTE DRIVE TO BE TESTED MUST BE LOADED WITH A CASSETTE WORK TAPE WHICH HAS THE WRITE PROTECT HOLES COVERED (WRITE ENABLED).

7. MODULE OPERATION

THE MODULE FIRST CHECKS IF THE LOAD MEDIUM IS CASSETTE. IF IT IS, THE LOAD DRIVE IS DROPPED FROM THE EXERCISE, SO AS NOT TO DESTROY CONTENTS OF LOAD CASSETTE. THE DRIVES ARE THEN REWOUND, AND THE FOLLOWING TEST SEQUENCE PERFORMED:

- A. SELECT A DRIVE
- B. WRITE A BLOCK OF 128 BYTES
- C. REVERSE ONE BLOCK
- D. READ BLOCK OF 128 BYTES
- E. CHECK THE READ DATA AGAINST WRITE DATA AND REPORT ERRORS.
(MAXIMUM OF 3 DATA ERRORS PER BLOCK ARE REPORTED).
- F. IF NOT 70 BLOCKS DONE, GOES TO STEP A.
- G. IF 70 BLOCKS DONE, REPORT END OF PASS, RESUME AT STEP A.

8. SPECIAL FEATURES

THE MODULE REMINDS A DRIVE WHENEVER IT DETECTS CLEAR LEADER, AND THEN REPEATS THE OPERATION ATTEMPTED WHEN CLEAR LEADER WAS DETECTED. (WRITE OR READ).

ALL SOFT ERRORS ARE REPORTED (TIMING ERROR, BLOCK CHECK, FILE GAP) HOWEVER, A WRITE OR READ COMMAND WILL BE TRIED UP TO 3 TIMES AFTER A SOFT ERROR. IF AFTER 3 TRIES THE PROGRAM DOES NOT SUCCEED IN PERFORMING THE OPERATION, AND ERROR IS INDICATED.

HARD ERRORS ARE REPORTED SEPARATELY FROM SOFT ERRORS (OFF-LINE, WRITE-LOCK). A DRIVE IS AUTOMATICALLY DROPPED FROM TEST AFTER A HARD ERROR. THE HARD ERROR IS REPORTED, THE FACT THAT THE DRIVE HAS BEEN DROPPED IS NOT. AFTER ALL DRIVES HAVE BEEN DROPPED, THE MODULE ITSELF IS DROPPED, SINCE THERE ARE NO DRIVES TO TEST. A MODULE DROPPED MESSAGE DOES OCCUR.

9. OPERATION OPTIONS:

4. MODULE LOCATION DVIDI (14) MAY BE CHANGED TO EXERCISE ANY COMBINATION OF UNITS 0-1. BIT0=DRV0, BIT1=DRV1.

5. IF DVIDI=0 AT RUNTIME THE MODULE WILL BE DROPPED FROM THE EXERCISE.

10. NON-STANDARD PRINTOUT

ALL PRINTOUTS HAVE STANDARD SIGNIFICANCE.

```

000000'      |
000000'      | IOMOD <TAAD >,177500,260,6,0,2,70.,7
000000'      | MODULE 140000,TAAD ,177500,260,6,0,2,70.,7
      | .TITLE TAAD DEC/X11 SYSTEM EXERCISER MODULE
      | DDXC0M VERSION 6 23-MAY-78
      | .LIST BIN
      | *****
000000'      | BEGIN:
000000' 040524 042101 040 MODNAM: .ASCII /TAAD /IMODULE NAME.
000005' 000 XFLAG: .BYTE OPEN ;USED TO KEEP TRACK OF WBUFF USAGE
000006' 177500 ADDR: 177500+0 ;1ST DEVICE ADDR.
000010' 040260 VECTOR: 260+0 ;1ST DEVICE VECTOR.
000012' 300 BR1: .BYTE PRTY6+0 ;1ST BR LEVEL.
000013' 000 BR2: .BYTE PRTY0+0 ;2ND BR LEVEL.
000014' 040003 DVIO1: 2+1 ;DEVICE INDICATOR 1.
000016' 000000 SR1: OPEN ;SWITCH REGISTER 1
000020' 000000 SR2: OPEN ;SWITCH REGISTER 2
000022' 000000 SR3: OPEN ;SWITCH REGISTER 3
000024' 000000 SR4: OPEN ;SWITCH REGISTER 4
      | *****
000026' 140000 STAT: 140000 ;STATUS WORD.
000030' 000250 INIT: START ;MODULE START ADDR.
000032' 000224 SPOINT: MODSP ;MODULE STACK POINTER.
000034' 000000 PASCNT: 0 ;PASS COUNTER.
000036' 000106 ICNT: 70. ;# OF ITERATIONS PER PASS=70.
000040' 000000 ICOUNT: 0 ;LOC TO COUNT ITERATIONS
000042' 000000 SOFCNT: 0 ;LOC TO SAVE TOTAL SOFT ERRORS
000044' 000000 HRDCNT: 0 ;LOC TO SAVE TOTAL HARD ERRORS
000046' 000000 SOFPAS: 0 ;LOC TO SAVE SOFT ERRORS PER PASS
000050' 000000 HRDPAS: 0 ;LOC TO SAVE HARD ERRORS PER PASS
000052' 000000 SYSCNT: 0 ;# OF SYS ERRORS ACCUMULATED
000054' 000000 RANNUM: 0 ;HOLDS RANDOM # WHEN RAND MACRO IS CALLED
000056' 000000 CONFIG:
000060' 000000 RES1: 0 ;RESERVED FOR MONITOR USE
000062' 000000 RES2: 0 ;RESERVED FOR MONITOR USE
000064' 000000 SVR0: OPEN ;LOC TO SAVE R0.
000066' 000000 SVR1: OPEN ;LOC TO SAVE R1.
000070' 000000 SVR2: OPEN ;LOC TO SAVE R2.
000072' 000000 SVR3: OPEN ;LOC TO SAVE R3.
000074' 000000 SVR4: OPEN ;LOC TO SAVE R4.
000076' 000000 SVR5: OPEN ;LOC TO SAVE R5.
000100' 000000 CSRA: OPEN ;LOC TO SAVE R6.
001102' SBADR: ;ADDR OF CURRENT CSR.
001104' ACSR: OPEN ;ADDR OF GOOD DATA, OR
001106' WASADR: ;ADDR OF BAD DATA, OR
001108' ASTAT: OPEN ;STATUS REG CONTENTS.
001110' ERRTYP: ;TYPE OF ERROR
001112' ASB: OPEN ;EXPECTED DATA.
001114' AWAS: OPEN ;ACTUAL DATA.
001116' RSTRT: RSTRT ;RESTART ADDRESS AFTER END OF PASS
001120' WOTO: OPEN ;WORDS TO MEMORY PER ITERATION
001122' WOFR: OPEN ;WORDS FROM MEMORY PER ITERATION
      | INTR: OPEN ;# OF INTERRUPTS PER ITERATION
      | IDNUM: 7 ;MODULE IDENTIFICATION NUMBER=7
    
```

```

000040'      | .REPT SPSIZ ;MODULE STACK STARTS HERE,
      | .NLST
      | .WORD 0
      | .LIST
      | .ENDR
000224'      | MODSP:
      | *****
    
```

```

210
211      000100      /FUNCTIONAL DEFINITION FOR TAI1
212      000400      INTRU=100
213      000001      UNIT1=400
214      000137      /SELECT UNIT 1
215      000105      GO=1
216      000103      FREWIND=16+INTRU+GO+20 /REWIND & GO
217      000131      FREAD=4+INTRU+GO /READ & GO
218      000040      FWRITE=2+INTRU+GO /WRITE & GO
219      000040      FBKSP=10+INTRU+GO+20 /BACKSPACE & GO
220      100000      /
221      020000      BDONE=40 /DONE
222      000020      BERROR=100000 /ERROR
223      000020      BCLRLDR=20000 /CLEAR LEADER
224      /PARAMETER BLOCK
225      /
226      000224' 000      /
227      000225' 000      /
228      000226' 000      /
229      000227' 000      /
230      /
231      000230' 000000      /
232      000231' 000000      /
233      000232' 000000      /
234      000233' 000000      /
235      000234' 000000      /
236      000235' 000000      /
237      000236' 000000      /
238      000237' 000000      /
239      000238' 000000      /
240      /
241      /REGISTER USAGE
242      /R0 =
243      /R1 =
244      /R2 =
245      /R3 =
246      /R4 =AVAILABLE UNIT STATUS
247      /R5 =CONTAINS TAI1 CSR ADDR.

```

```

247      /MODULE CODE STARTS HERE
248      /
249      000250' 012767 000100 177636 /START: MOV #64,WDTO /64 WORDS TO MEM PER ITERATION
250      000256' 012767 000100 177632 /MOV #64,WDFR /64 WORDS FROM MEM PER ITERATION
251      000264' 012767 000405 177626 /MOV #261,INTR /261 INTERRUPTS PER ITERATION
252      000272' 016705 177510 /MOV ADDR,R5 /GET TAI1 ADDRESS.
253      000276' 010567 177576 /MOV R5,CSRA /GET TAI1 ADDR TO CSRA NOW.
254      000302' 010567 177740 /MOV R5,TTADB /SET UP TADB ADDR.
255      000306' 062767 000002 177732 /ADD #2,TTADB
256      000314' 016700 177470 /MOV VECTOR,R0 /GET THE VECTOR ADDRESS
257      000320' 012720 001052' /MOV #TAINTR,(0)+ /POINT THE INTERRUPTS TO TAINTR
258      000324' 016720 177462 /MOV BR1,(0)+ /SET UP THE PRIORITY
259      000330' 016704 177460 /MOV DVID1,R4 /SAVE DEVICE SELECTION PARAMETERS
260      000334' 122737 000005 000041 /CMPB #5,#41 /LOAD MEDIUM CASSETTE?
261      000342' 001011 /BNE 13 /BR IF NOT
262      000344' 123727 000040 000000 /CMPB #40,#0 /LOAD UNIT 0?
263      000352' 001003 /BNE 30 /FNO, UNIT 1
264      000354' 042704 000001 /BIC #1,R4 /DON'T TEST UNIT 0
265      000364' 004002 /BR 13
266      000362' 042704 000002 /BIC #2,R4 /DON'T TEST UNIT 1
267      /REWIND BOTH UNITS
268      000366' 004767 000112 /JBR PC,SEODRV /SELECT UNIT
269      000372' 004767 000154 /JBR PC,REWIND /REWIND IT.
270      000376' 000240 /NOP /ERROR RETURN.
271      000400' 004767 000100 /JBR PC,SEODRV /SELECT OTHER UNIT.
272      000404' 004767 000142 /JBR PC,REWIND /REWIND IT TOO.
273      000410' 000240 /NOP /ERROR RETURN.
274      /
275      /
276      000412' 012701 000040 /LOAD WRITE BUFFER AREA
277      000416' 012700 001414' /MOV #32,R1 /SET COUNTER
278      000422' 012720 125125 /MOV #WTFUP,R0 /GET WRITE BUFFER ADDRESS
279      000426' 012720 000377 /MOV #125125,(R0)+ /LOAD WRITE BUFFER AREA
280      000432' 005301 /DEC R1 /
281      000434' 001372 /BNE R1 /DONE ?
282      /NO, KEEP LOADING
283      /
284      000436' 012767 001414' 177564 /TEST SEQUENCE
285      000444' 004767 000034 /RESTR: MOV #WTFUP,WTFUP /GET WRITE BUFFER ADDR.
286      000450' 004767 000202 /JBR PC,SEODRV /SELECT A DRIVE.
287      000454' 000773 /JBR PC,WRITE /WRITE A BLOCK.
288      000456' 004767 000132 /BR 13 /ERROR RETURN.
289      000462' 000770 /JBR PC,REVLK /BACKSPACE A BLOCK.
290      000464' 004767 000204 /BR 13 /ERROR RETURN.
291      000470' 000765 /JBR PC,READ /READ THE BLOCK.
292      000472' 004767 000630 /BR 13 /ERROR RETURN.
293      000476' 104413 000000' /JBR PC,CKDAT /CHECK THE DATA.
294      /ENDITS,BEGIN /SIGNAL END OF ITERATION.
295      000502' 000760 /BR 13 /MONITOR SHALL TEST END OF PASS

```

```

296
297 000504 012700 000002          ;ROUTINE TO SELECT A DRIVE.
298 000510 105167 177513          SEQDRV: MOV #2,R0          ;WILL TRY TWICE.
299 000514 142767 000376 177505 13: COMB UNIT          ;TRY OTHER UNIT.
300 000522 001407                   BICB #376,UNIT          ;NEED ONE BIT ONLY.
301 000524 032704 000002          BEQ 33                   ;BR IF 0 (UNIT 0).
302 000530 001007                   BIT #BIT1,R4           ;UNIT 1 AVAILABLE?
303 000532 005300                   BNE 43                   ;BR IF YES.
304 000534 001365                   DEC R0                   ;TRIED TWICE?
305 000536 104410 000000          BNE 13                   ;BR IF NOT.
306 000542 032704 000001          ENDS,BEGIN             ;NO. DROP MODULE. NO DRIVES AVAILABLE.
307 000546 001771                   BIT #BIT0,R4           ;UNIT 0 AVAILABLE?
308 000550 000207                   BEQ 23                   ;BR IF NOT.
309                                     RTS PC                   ;EXIT, GOT A DRIVE.
310
311 000552 012767 000576 177454          ;ROUTINE TO REWIND SELECTED DRIVE.
312 000560 112767 000137 177440          REWIND: MOV #13,FORK          ;
313 000566 016715 177434                   MOVB #PREWIND,CMND      ;SET UP REWIND COMMAND IN COMMAND WORD.
314 000572 104400 000000          MOV CMND,(5)           ;ISSUE COMMAND.
315 000576 004767 000342          EXITS,BEGIN           ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
316 000602 000240                   JSR PC,CKER           ;CHECK FOR ERRORS.
317 000604 000402                   NOP                      ;HARD ERROR.
318 000606 062716 000002          BR 23                   ;SOFT ERROR.
319 000612 000207                   ADD #2,(6)              ;OK RETURN.
320                                     RTS PC                   ;DONE, EXIT.
321
322 000614 012767 000640 177412          ;ROUTINE TO BACKSPACE A BLOCK.
323 000622 112767 000131 177376          REVBLK: MOV #13,FORK          ;
324 000630 016715 177372                   MOVB #FBKSP,CMND       ;SET UP REV BLK COMMAND IN COMMAND WORD.
325 000634 104400 000000          MOV CMND,(5)           ;ISSUE COMMAND.
326 000640 004767 000300          EXITS,BEGIN           ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
327 000644 000240                   JSR PC,CKER           ;CHECK FOR ERRORS.
328 000646 000402                   NOP                      ;HARD ERROR.
329 000650 062716 000002          BR 23                   ;SOFT ERROR.
330 000654 000207                   ADD #2,(6)              ;OK RETURN.
331                                     RTS PC                   ;DONE, EXIT.
332

```

```

333
334 000656 112767 000103 177341          ;READ/WRITE ROUTINE.
335 000664 016767 177340 177340          WRITE: MOVB #WRITE,RWCMND    ;SET UP WRITE COMMAND IN COMMAND WORD.
336 000672 000406                   MOV TMBUF,BUF          ;
337 000674 112767 000105 177323          BR RWCMND              ;
338 000702 012767 001614 177322          READ: MOVB #READ,RWCMND    ;SET UP READ COMMAND IN COMMAND WORD.
339 000710 012767 000003 177324          MOV #RD0BUF,BUF       ;
340 000714 016767 177310 177312          RWCOM: MOV #3,TRYCTR      ;WILL TRY 3 TIMES.
341 000720 012767 000756 177302          13: MOV BUF,TMPBUF      ;LOAD TEMP BUFF.
342 000732 012767 000200 177300          MOV #23,FORK          ;POINT FORK TO 23.
343 000740 116767 177261 177260          MOV #128,BYTCNT       ;WILL READ/WRITE 128 BYTES.
344 000746 016715 177254                   MOVB RWCMND,CMND       ;MOVE RW COMMAND TO COMMAND WORD.
345 000752 104400 000000          MOV CMND,(5)           ;ISSUE COMMAND.
346 000756 004767 000162          EXITS,BEGIN           ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
347 000762 000403                   JSR PC,CKER           ;CHECK FOR ERRORS.
348 000764 000403                   BR 33                   ;HARD ERROR.
349 000766 062716 000002          BR 43                   ;SOFT ERROR.
350 000772 000207                   ADD #2,(6)              ;OK RETURN.
351 000774 032767 020000 177242          43: RTS PC              ;DONE, EXIT.
352 001002 001404                   BIT #20000,TMPCSR      ;WAS IT CLEAR LEADER?
353 001004 004767 177542                   BEQ 53                   ;BR IF NOT.
354 001010 000770                   JSR PC,REWIND          ;REWIND CASSETTE.
355 001012 000741                   BR 33                   ;ERROR RETURN.
356 001014 005367 177222                   BR 13                   ;
357 001020 001404                   DEC TRYCTR             ;TRIED 3 TIMES?
358 001022 004767 177566                   BEQ 63                   ;BR IF YES.
359 001026 000761                   JSR PC,REVBLK         ;NO, REVERSE BLOCK.
360 001030 000732                   BR 33                   ;ERROR RETURN.
361 001032 004767 000252                   BR 13                   ;GO READ AGAIN.
362 001036 005067 177044                   JSR PC,ERRSET         ;COMMON ERROR SETUP.
363                                     CLR ERRTP              ;UNKNOWN ERROR CODE.
364
365 001042 104405 000000 000000          ;*****
366 001050 000207                   HRDERS,BEGIN,NULL     ;FAILED TO READ/WRITE WITHIN 3 TRIES
367                                     ;*****
368                                     RTS PC                 ;ERROR RETURN.
369
370 001052 105777 176730                   ;CASSETTE INTERRUPT HANDLER
371 001056 100405          TAINTR: T3YB #ADDR          ;TRANSFER REQUEST?
372                                     BMI TREQ              ;YES = PIRQ 13
373                                     ;-----
374 001060 000004 000000 001066          PIRQS,BEGIN,13        ;QUEUE UP TO CONTINUE AT 13 AND RTI
375                                     ;-----
376 001066 000177 177142          13: JMP #FORK              ;GO TO SERVICE CURRENT SEQUENCE
377 001072 005367 177142          TREQ: DEC BYTCNT         ;WHEN COUNT NEG., DONE
378 001076 100416                   BMI 43                 ;GO TO LAST BYTE SEQUENCE
379 001100 032777 000004 176700          BIT #4,#ADDR          ;CHECK READ OR WRITE FUNCTION
380 001106 001004                   BNE 13                 ;ONE = GO TO PROCESS READ
381 001110 117777 177122 177130          MOVB #TMPBUF,#TTADB   ;WRITE BYTE
382 001116 000403                   BR 23                   ;
383 001120 117777 177122 177110          13: MOVB #TTADB,#TMPBUF  ;READ BYTE
384 001126 005267 177104          23: INC TMPBUF          ;UPDATE ADDR.
385 001132 000002                   RTI                    ;RETURN FROM INT.
386 001134 052777 000020 176644          43: BIS #20,#ADDR       ;INITIATE LAST BYTE SEQUENCE
387 001142 000002                   RTI                    ;EXIT INTERRUPT SEQUENCE.

```

```

386                                     ;ROUTINE TO CHECK FOR ERRORS.
387 001144 032715 000040 CKER: BIT #40,(5) ;DONE BIT SET?
388 001150 001011 PNE 13 ;BR IF YES.
389 001152 004767 000132 JSR PC,ERRSET ;NO, ERROR SETUP.
390 001156 012767 000003 176722 MOV #3,ERRTYP ;CONTROLLER NOT READY
391 ;*****
392 001164 104405 000000 000000 HRDERS,BEGIN,NULL ;DONE/XFR REQUEST NOT SET
393 ;*****
394 001172 000424 18: TST (5) ;ANY ERRORS?
395 001174 005715 BPL 38 ;BR IF NOT.
396 001176 100020 JSR PC,ERRSET ;DO ERROR SETUP.
397 001200 004767 000104 BIT #1000,TMPCSR ;WAS IT HARD ERROR?
398 001204 032767 011000 177032 BNE 65 ;BR IF YES.
399 001212 001017 020000 177022 BIT #20000,TMPCSR ;CLEAR LEADER?
400 001214 032767 020000 177022 BNE 48 ;BR IF YES, DO NOT REPORT HARD ERROR.
401 001222 001010 CLR ERRTYP ;UNKNOWN SOFT ERROR.
402 001224 005067 176656 ;COULD BE TIMING,BLOCK,ETC
403 ;*****
404 001230 104406 000000 000000 ;*****
405 001230 104406 000000 000000 ;*****
406 001236 000402 BR 48 ;*****
407 001240 062716 000002 38: ADD #2,(6) ;OK RETURN SETUP.
408 001244 062716 000002 48: ADD #2,(6) ;SOFT ERROR RETURN SET UP.
409 001250 000207 58: RTS PC ;RETURN TO SEQUENCE
410 001252 012767 000006 176626 68: MOV #6,ERRTYP ;OFF LINE OR WRITE PROTECTED
411 ;*****
412 001260 104405 000000 000000 HRDERS,BEGIN,NULL ;*****
413 001260 104405 000000 000000 ;*****
414 001266 105767 176735 TSTB UNIT ;DETERMINE UNIT NUMBER.
415 001272 001003 BNE 75 ;BR IF UNIT 1.
416 001274 042704 000001 BIC #BIT0,R4 ;DROP UNIT 0.
417 001300 000763 BR 53
418 001302 042704 000002 78: BIC #BIT1,R4 ;DROP UNIT 1
419 001306 000760 BR 53
420 001310 011567 176566 ERRSET: MOV (5),ACSR ;SAVE CSR CONTENTS IN ACSR,
421 001314 011567 176724 MOV (5),TMPCSR ;AND TMPCSR,
422 001320 012715 000020 MOV #BIL05,(5) ;STOP MOTION,
423 001324 000267 RTS PC ;EXIT.
    
```

```

426                                     ;DATA CHECK ROUTINE. REPORTS UP TO 3 ERRORS PER BLOCK CHECKED.
427 001326 016700 176676 CKDAT: MOV TMBUF,R0 ;WRITE BUFFER ADDR TO R0.
428 001332 012701 001614 MOV #RDBUF,R1 ;READ BUFFER ADDR TO R1.
429 001336 012702 000200 MOV #128,R2 ;WILL CHECK 128 BYTES.
430 001342 012703 000003 MOV #3,R3 ;WILL REPORT UP TO 3 ERRORS.
431 001346 121011 18: CMPB (0),(1) ;CHECK A SET OF BYTES.
432 001350 001414 BEQ 25 ;BR IF SAME, GOOD.
433 001352 010067 176524 MOV R0,SBADR ;SAVE GOOD ADDR.
434 001356 010167 176522 MOV R1,WASADR ;SAVE BAD ADDR.
435 001362 111067 176520 MOVB (#),ASB ;SAVE GOOD BYTE.
436 001366 111167 176516 MOVB (1),AWAS ;SAVE BAD BYTE.
437 ;*****
438 001372 104404 000000 000000 ;*****
439 001376 005303 DATERS,BEGIN ;DATA ERROR!!!
440 001400 001404 DEC R3 ;3 ERRORS REPORTED?
441 001402 005302 BEQ 38 ;BR IF YES, QUIT.
442 001404 001402 28: DEC R2 ;ALL BYTES CHECKED?
443 001406 122021 BEQ 33 ;BR IF YES, QUIT.
444 001410 000756 CMPB (0)+,(1)+ ;UPDATE BYTE ADDRESSES.
445 001412 000207 BR 13 ;DO IT AGAIN.
446 001414 000207 38: RTS PC ;EXIT.
447
448 001414 000200 WYBUF: .BLKB 120. ;WRITE BUFFER AREA
449 001614 000200 RDBUF: .BLKB 120. ;INPUT BUFFER AREA
450 000001 .END
    
```


SR2	000020R	164#			
SR3	000022R	165#			
SR4	000024R	166#			
START	000250R	169	249#		
STAT	000026R	168#			
SVR0	000062R	183#			
SVR1	000064R	184#			
SVR2	000066R	185#			
SVR3	000070R	186#			
SVR4	000072R	187#			
SVR5	000074R	188#			
SVR6	000076R	189#			
SYSCNT	000052R	178#			
TAINTR	001052R	257	369#		
THPBUF	000236R	234#	340#	379	381#
THPCOR	000244R	237#	351	398	400
TREQ	001072R	370	375#		423#
TRPDFD	000022	218#			
TRYCTR	000242R	236#	339#	356#	
TTADB	000246R	238#	254#	255#	379#
THBUF	000230R	231#	284#	335	427
UNIT	000227R	229#	298#	299#	415
UNIT1	000400	212#			
VECTOR	000010R	159#	256		
WASADR	000104R	193#	434#		
WDFR	000116R	200#	250#		
WOTO	000114R	199#	249#		
WRITE	000656R	286	334#		
WTBUF	001414R	277	284	448#	
XFLAG	000008R	157#			
.	002014R	448#	449#		

. ABS. 000000 000
002014 001

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

XTAAD0,XTAAD0/SOL/CRF:SYM=DDXCOM,XTAAD0
RUN=TIME: 1 1 ,2 SECONDS
RUN=TIME RATIO: 12/3=3,8
CORE USED: 7K (13 PAGES)

END USER DAVIES,TOM [400,2704] JOB XTAAD0 SEQ. 7128 DATE 12=DEC=78 15:25:41 MONITOR IPC=F 603 [7A0] **END**

END USER DAVIES,TOM [400,2704] JOB XTAAD0 SEQ. 7128 DATE 12=DEC=78 15:25:41 MONITOR IPC=F 603 [7A0] **END**

END USER DAVIES,TOM [400,2704] JOB XTAAD0 SEQ. 7128 DATE 12=DEC=78 15:25:41 MONITOR IPC=F 603 [7A0] **END**

*** L P T S P L R U N L O G ***

15:24:07 LPDAT [LPTLSJ LPTSPL VERSION 102(2263)/3(61) RUNNING ON LPT630, 12=DEC=78 15:24:07]
15:24:07 LPDAT [LPTSJS STARTING JOB XTAAD0, SEQ #7128, REQUEST CREATED AT 12=DEC=78 15:24:07]
15:24:16 LPM86 [LPTSFP STARTING FILE DSKZ:XTAAD0,SEQ<057>[400,2704]]
15:25:41 LPM86 [LPTFPF FINISHED PRINTING FILE DSKZ:XTAAD0,SEQ<057>[400,2704]]
15:25:41 LPSUM SPOOLER RUNTIME 1 SECONDS, 15 KCS, 47 DISK READS, 15 PAGES PRINTED

/TO:ML21=4:DAVIES == DISTRIBUTION TO ML21=4, SLOT 134

END USER DAVIES,TOM [400,2704] JOB XTAAD0 SEQ. 7128 DATE 12=DEC=78 15:25:41 MONITOR IPC=F 603 [7A0] **END**

/TO:ML21=4:DAVIES == DISTRIBUTION TO ML21=4, SLOT 134

END USER DAVIES,TOM [400,2704] JOB XTAAD0 SEQ. 7128 DATE 12=DEC=78 15:25:41 MONITOR IPC=F 603 [7A0] **END**

/TO:ML21=4:DAVIES == DISTRIBUTION TO ML21=4, SLOT 134

END USER DAVIES,TOM [400,2704] JOB XTAAD0 SEQ. 7128 DATE 12=DEC=78 15:25:41 MONITOR IPC=F 603 [7A0] **END**

/TO:ML21=4:DAVIES == DISTRIBUTION TO ML21=4, SLOT 134

END USER DAVIES,TOM [400,2704] JOB XTAAD0 SEQ. 7128 DATE 12=DEC=78 15:25:41 MONITOR IPC=F 603 [7A0] **END**

/TO:ML21=4:DAVIES == DISTRIBUTION TO ML21=4, SLOT 134

END USER DAVIES,TOM [400,2704] JOB XTAAD0 SEQ. 7128 DATE 12=DEC=78 15:25:41 MONITOR IPC=F 603 [7A0] **END**